

# Testausdokumentti

Projekti: Splash

Tekijä: Sami-Pekka Ollila

Tekniikka: Maven ja PIT

## Yleisrakenne

Ohjelma suoritus hyödyntää kahta eri algoritmi tyyppiä, valinta algoritmi (Selection algorithm) joka valitsee alueen käyttäjän klikkauksen perusteella sekä värin korvaus (Replacing algorithm) joka häivyttää valitun alueen pois.

Ohjelman muut luokat / tietorakenteet / oliot ovat olemassa, jotta edellä mainitut algoritmit pääsisivät käsiksi pikseli dataan ja käyttöliittymän kannalta.

## Aika ja tilavaativuudet

### Selection algoritmit:

- DummyLoop:
  - Yksinkertainen algoritmi, joka käy kaikki kuvan pikselit läpi ja merkitsee pikselin valituksi, mikäli sen värin erotus käyttäjän valitsemaan aloituspisteeseen on suurempi kuin parametrina määritelty arvo "range".
  - **Aika:** Algoritmi käy kaikki kuvan pikselit kerran läpi, joten  $O(n)$ .
  - **Tila:** Algoritmi tallentaa valitut pikselit matriisiin jonka koko on sama kuin kuvan koko, eli  $O(n)$ .
- Flood Fill:
  - Flood Fill algoritmi pyrkii löytämään valitun alueen vasemman ja oikean reunan, jonka jälkeen kaikki reunojen välillä olevat pikselit valitaan. Pikseleiden vertailu on toteutettu kuten DummyLoop:issa.
  - **Aika:** Pahimmassa tapauksessa kaikki pikselit käydään kerran läpi, joten  $O(n)$ .
  - **Tila:** Algoritmi tallentaa valitut pikselit matriisiin jonka koko on sama kuin kuvan koko, eli  $O(n)$ .
- OnePass Connected-component labeling :
  - Algoritmi käy kaikki pikselit läpi ja pyrkii niiden värin perusteella luomaan samanvärisistä pikseleistä ryhmiä. Algoritmi ei toiminut tätä tehtävää varten kovin hyvin, sillä se on

huomattavasti hitaampi kuin muut, mutta jos esimerkiksi valittua aluetta pitäisi muuttaa usein saattaisi tämä olla toimiva ratkaisu.

- **Aika:** Pahimmassa tapauksessa kaikki pikselit käydään kerran läpi, joten  $O(n)$ .
- **Tila:** Algoritmi tallentaa valitut pikselit matriisiin jonka koko on sama kuin kuvan koko, eli  $O(n)$ .

### Replace Selectionit:

- DummyReplace:
  - Yksinkertainen algoritmi, joka käy kaikki valitut pikselit läpi ja vaihtaa niiden väriksi seuraavan valitsemattoman pikselin värin joko oikealta tai vasemmalta (riippuen siitä kumpi eroaa enemmän valitusta pikselistä).
  - **Aika:** Algoritmi käy kaikki kuvan pikselit kerran läpi, joten  $O(n)$ .
  - **Tila:** Algoritmi ei vaadi muistia muuhun kuin väliaikaisten pikseleiden tallentamiseen, joten tilavaativuus on vakio.
- Vertical Repeat:
  - Vertical Repeat kopioi valitun alueen vasemmalta puolelta saman kokoisen alueen kuin on valittu, peilaa sen valitun alueen päälle korvaten valitut pikselit. Toteutus ei toimi kovin hyvin, jos valinta tehdään kuvan vasempaan laitaan tai jos tausta ei ole yksivärinen.
  - **Aika:** Pahimmassa tapauksessa kaikki pikselit käydään kerran läpi, joten  $O(n)$ .
  - **Tila:** Tilaa tarvitaan valitun alueen verran, eli  $O(n)$ .
- Mixed Repeat :
  - Algoritmi toimii samalla periaatteella kuin Vertical Repeat, mutta kopioi alueen valitun alueen vasemmalta, oikealta, ylä- ja alapuolelta. Lopuksi kaikkien neljän eri suunnan pikseleiden keskiarvoa tutkitaan ja pyritään päättämään minkä värinen pikseli sopisi parhaiten tietyn pikselin korvaamiseen. Algoritmille riittää toimiakseen, että se onnistuu kopioimaan alueen edes yhdestä ilmansuunnasta.
  - **Aika:** Algoritmi käy pikselit lävitse kaksi kertaa, ensimmäisellä kerralla etsitään alueen reunat ja toisella asetetaan pikseleiden värit.  $O(2n)$ .
  - **Tila:** Algoritmi tallentaa kaikille eri ilmansuunnille omat matriisinsa, joten  $O(4n)$ .

### Testaus

Tällä hetkellä kaikki algoritmit on testattu unit testeillä joilla varmistetaan, että ne hoitavat niille asetetut kriteerit. Kun suorituskyky testaukset on tehty, valitaan paras Selection ja Replace algoritmi jatkotestaamista varten ja niiden testikattavuutta lisätään.

**Lähteet:**

[https://en.wikipedia.org/wiki/Flood\\_fill](https://en.wikipedia.org/wiki/Flood_fill)

[https://en.wikipedia.org/wiki/Connected-component\\_labeling](https://en.wikipedia.org/wiki/Connected-component_labeling)

[https://en.wikipedia.org/wiki/Blob\\_detection](https://en.wikipedia.org/wiki/Blob_detection)

<https://en.wikipedia.org/wiki/Inpainting>

<http://www.dtic.upf.edu/~mbertalmio/bertalmi.pdf>